

## Contents

- 1 Tutorial: Hinzufügen eines SD-Kartenlesers an den Linksys WRT54G
  - ◆ 1.1 Bilder
  - ◆ 1.2 Was man braucht
  - ◆ 1.3 Wie man vorgeht
    - ◇ 1.3.1 1 - Einleitung
      - 1.3.1.1 GPIO Pins, hä?
        - 1.3.1.1.1 Die Pins bei einem WRT54G v2
        - 1.3.1.1.2 Die Pins bei einem WRT54GL v1.1
    - ◇ 1.3.2 2 - Achtung!
    - ◇ 1.3.3 3 - SD Kartenleser anschliessen
    - ◇ 1.3.4 4 - SD Kartenleser befestigen
    - ◇ 1.3.5 5 - Fertig
  - ◆ 1.4 Software
    - ◇ 1.4.1 Aktuelle Methode
    - ◇ 1.4.2 Alte Methode
      - 1.4.2.1 Etwas Hilfe beim Kernel kompilieren
    - ◇ 1.4.3 Welche SD Karten funktionieren?
    - ◇ 1.4.4 SD Karten die nicht funktionieren!
  - ◆ 1.5 Fehlerbehebung
- 2 Version herausfinden
- 3 Informationen über Hardware
- 4 Overclocking der CPU

## Tutorial: Hinzufügen eines SD-Kartenlesers an den Linksys WRT54G

*von Rasmus Rohde und Mads Ulrik Kristoffersen - übersetzt ins Deutsche von gammelgemse*

Dieses Projekt ist für diejenigen gedacht die gern ihren WRT54G um etwas mehr Speicher ergänzen würden, zusätzlich zu den 4 MB eingebauten Flash-RAM. Was wir tun werden ist, ein SD-Kartenleser an einigen GPIO-Pins, die sich im Inneren an der CPU befinden, anzuschließen. Mittels eines kleinen Treibers kann man ihn dann unter Linux nutzen. Das heißt, wenn man seinen Kernel für den Linksys beispielsweise mit Unterstützung für MS-DOS Partitionen und VFAT kompiliert, wird es möglich sein, den Speicher der jeweiligen SD-Karte als lese- und schreibbare Partition zu mounten. Die erreichbare Schreib-/ Lesegeschwindigkeit liegt bei etwa 200 KB/s.

## Bilder

- ◆ Das Innenleben eines Routers mit eingebautem SD-Kartenleser  
( <http://kiel.kool.dk/pics/AllSolderingDone.jpg> )

- ◆ Der fertige Router mit installiertem SD-Kartenleser  
( [http://kiel.kool.dk/pics/Reuter\\_complete.jpg](http://kiel.kool.dk/pics/Reuter_complete.jpg) ) und  
( [http://www.duff.dk/wrt54gs/pics/Reuter\\_complete.jpg](http://www.duff.dk/wrt54gs/pics/Reuter_complete.jpg) )
- ◆ Das Innenleben mit und ohne eingebautem SD-Kartenleser  
(<http://www.adamkowalewski.com>)

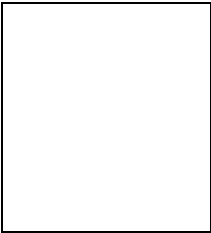
## Was man braucht

- ◆ ein LötKolben und etwas LötZinn (und etwas LötErfahrung)
- ◆ ein SD-Kartenleser, es sei denn man lötet die SD-Karte direkt an <- oder ein altes Floppy-Kabel mit 5,25 Zoll Stecker! (Bauanleitung Floppy-SD-card-Reader)
- ◆ 6 kleine Drahtstücke (kurz!!! niemals länger als 20cm, am besten nur 5-10cm!)
- ◆ ein Linksys WRT54G (Hardware Revision < v5)

## Wie man vorgeht

### 1 - Einleitung

Damit die SD-Karte funktioniert, müssen 6 Drähte angelötet werden. Folgendes Bild sollte eine Übersicht über die Pins geben, die dafür in Frage kommen.

	<ol style="list-style-type: none"><li>1. CS - Chip-Select Leitung</li><li>2. DI - Data-In Leitung zur SD-Karte</li><li>3. VSS - Masse 1</li><li>4. VDD - 3,3-V-Leitung für die Stromversorgung</li><li>5. CLK - Taktleitung für die SD-Karte</li><li>6. VSS2 - Masse 2</li><li>7. DO - Data-Out-Leitung von der SD-Karte</li></ol>	GPIO 7 GPIO 5 / GPIO 2  GPIO 3  GPIO 4
---	--	--

Die SD-Karte wird im SPI-Mode betrieben, das heißt dass nur einer (Pin 7) der 4 Data-Out-Pins zur Datenübertragung genutzt wird. Spezifikationen, um die Karte im nativen SD-Mode zu betreiben sind SEHR kostenaufwändig zu beschaffen und außerdem zwingt die eingeschränkte Anzahl an GPIO-Pins, die im Router verfügbar sind, zur Nutzung eines seriellen Protokolls. Die zwei VSS-Pins können in diesem Fall einfach zusammengelötet werden (VSS2 kontrolliert den Sleep-Mode der Karte). Das berücksichtigend betrachten wir weiter die einzelnen Lötstellen im Router.

## Hinzufügen\_eines\_SD-Kartenlesers\_an\_den\_Linksys\_WRT54G

### GPIO Pins, hä?

Die integrierte Broadcom-BCM4712-CPU stellt eine Reihe von General Purpose Input/Output Pins (oder GPIO-Pins) zur Verfügung, die für verschiedene Zwecke im Router genutzt werden. Wir konnten bislang 8 dieser Pins identifizieren, die die folgenden zugewiesenen Funktionen haben:

Pin	Richtung	Name
GPIO 0	(Output)	WLAN LED
GPIO 1	Output	Power LED
GPIO 2	Output	ADM_EECS
GPIO 3	Output	ADM_EESK
GPIO 4	Input	ADM_EEDO
GPIO 5	Output	ADM_EEDI
GPIO 6	Input	Reset button
GPIO 7	Output	DMZ LED

Die Pins, die in diesem Projekt genutzt werden, sind ADM\_EESK, ADM\_EEDO, ADM\_EEDI und die DMZ-LED. Die ADM\_\*-Pins stellen normalerweise ein Interface dar, um den ADMtek-Switch-Chip zu konfigurieren. Da dies aber nur während des Boot-Vorgangs passiert, können wir die Pins danach für unsere Zwecke nutzen (die entsprechenden Pins am Switch-Chip sind danach hochohmig geschaltet). Die Bezeichnungen der anderen Pins sollten selbsterklärend sein. Die Richtung der Pins kann individuell programmiert werden (selbst wenn das nicht für jeden einzelnen Pin Sinn macht).

### Die Pins bei einem WRT54G v2

die ersten drei Lötstellen liegen bei RP3  
die nächsten zwei Lötstellen sind bei JP1  
die letzte Lötstelle ist an der DMZ-LED

### Die Pins bei einem WRT54GL v1.1

Bilder von CaScAdE:

<http://cascade.dyndns.org/~datagarbage/linksys-wrt54gl-v1.1-3.3v+GND.jpg>

<http://cascade.dyndns.org/~datagarbage/linksys-wrt54gl-v1.1-gpio-2+3.jpg>

<http://cascade.dyndns.org/~datagarbage/linksys-wrt54gl-v1.1-gpio-4+7.jpg>



- Bilder sind down, links werden gelöscht wenn sie nicht ersetzt werden!

## Hinzufügen\_eines\_SD-Kartenlesers\_an\_den\_Linksys\_WRT54G

- Bilder von Meex:
- <http://home.arcor.de/maxigolla/.wrt54gl/linksys-wrt54gl-v1.1-3.3v+GND.jpg>
- <http://home.arcor.de/maxigolla/.wrt54gl/GPIO2.jpg>
- <http://home.arcor.de/maxigolla/.wrt54gl/GPIO3.jpg>
- <http://home.arcor.de/maxigolla/.wrt54gl/linksys-wrt54gl-v1.1-gpio-4+7.jpg>
- 



## 2 - Achtung!

Besonders ist darauf zu achten die RP3 Pins nicht kurzzuschließen, denn obwohl diese Lötstellen ausgesucht wurden, weil die den meisten Platz um die betreffenden GPIO-Leitungen beim Löten bieten, ist es trotzdem ziemlich eng - also aufpassen!

## 3 - SD Kartenleser anschliessen

Nachdem die Leitungen auf der Routerseite sauber angelötet sind, werden sie als nächstes mit der

Dieses Bild zeigt den SD-Kartenleser. Daran kann man sie besonders einfach anlöten.

Andere Möglichkeit : (Geschrieben von hollari, Idee stammt von irgendwo aus dem I-NET, Quelle wird nachgereicht, falls ich die noch finde)

Man verwendet den 5,25 Zoll Stecker, wie er normalerweise für Diskettenlaufwerke verwendet wird.

Dazu entfernt man als erstes die Halterung für das Flachbandkabel.

Je nach Typ braucht man nur daran zu ziehen, oder es ist ein Widerhaken vorhanden, den man mit einem kleinen Schraubendreher zur Seite drückt.

Danach zieht man vorsichtig das Flachbandkabel ab. Unter Zuhilfenahme einer Speicherkarte kann man nun prüfen, welche Pins benötigt werden.

Auf einer Seite entfernt man nun ALLE Federn mit einer kleinen Zange.

Auf der anderen Seite entfernt man die erste, lässt 7 Stk. drin, und entfernt die restlichen auch noch.

Dadurch ist sichergestellt, dass die Karte leicht rein- und rausgeht.

Nun schneidet man vom Kabel ein ausreichend langes Stück ab (20 cm?), und zieht mit den Fingernägeln ein Kabel mit 15 Pins ab. Mit etwas Kraft kann man das Kabel in den Stecker wieder einpressen.

Dabei darauf achten, dass vor- und nach den benötigten 7 Pins noch ein zusätzliches Kabelchen frei bleibt.

Die Pins bei einem WRT54GL v1.1

## Hinzufügen\_eines\_SD-Kartenlesers\_an\_den\_Linksys\_WRT54G

Die werden später an Masse angeschlossen, und sollten eine ausreichende Schirmung der Datenleitungen untereinander bieten.

Als letztes zieht man am anderen Ende des 15pin-Flachbandkabels alle Adern ca. 5cm auseinander, und lötet sie an der Platine des WRT54G fest. Wie vorher erwähnt kommt jeder 2. Pin an Masse :

Masse - GPIO4 - Masse - Masse - Masse - GPIO3 - Masse - +3,3V - Masse - Masse - Masse - GPIO2/5 - Masse - GPIO7 - Masse

Das Flachbandkabel kann man dann ganz einfach durch den grossen Schlitz (für die 4xLAN-Anschlüsse) nach draussen führen. Es ist dann automatisch eingeklemmt, und bietet so eine gewisse Zugentlastung.

Wer möchte, kann den Kartenleser natürlich auch innen wo einbauen.

Der Nachteil hierbei ist leider, dass man mit der Karte relativ genau "zielen" muss, damit man alle Kontakte trifft. Ausserdem darf die Karte NICHT bis zum Anschlag eingesteckt werden, sonst -> Kein Kontakt !

### 4 - SD Kartenleser befestigen

Befestige den SD-Kartenleser irgendwo innerhalb des Gehäuses. Wir haben uns für die rechte Gehäuse-Vorderseite entschieden und benutzten doppelseitiges Klebeband zum Befestigen und bohrten einen schmalen Schlitz ins Gehäuse, um SD-Karten bei geschlossenem Gehäuse bequem von aussen ein- und ausstecken zu können. Auf den Bilder-Links am Anfang der Seite sieht man wie es aussieht, ebenso auf diesem Bild vom Schlitz.

### 5 - Fertig

So weit, so einfach. Jetzt folgt der Software-Teil.

## Software

### Aktuelle Methode

Bei DD-WRT in aktuellen Versionen ist KEIN neu kompilieren des Kernels notwendig, es reicht unter Administration -> Managment -> MMC/SD Card Support den Haken auf enabled zu setzen:

<http://cascade.dyndns.org/~datagarbage/dd-wrt-enable-mmc.png>

Beim ersten start wird, sofern nicht vorhanden, ein ext2 (NICHT vfat!!!!) Dateisystem erzeugt, dies kann einige Zeit dauern, aber nur wens noch nicht vorhanden ist, also keine Angst, in zukunft startet der Router wieder schneller.

Aber Achtung: Oftmals funktioniert genau das nicht. Es ist unabdingbar, das eine Partition auf der Karte vorhanden ist.

Wenn das nicht der Fall ist, kommt genau so eine MMC-INIT-Error, als wenn der SD-Mod fehlgeschlagen ist.

## Hinzufügen\_eines\_SD-Kartenlesers\_an\_den\_Linksys\_WRT54G

Das führt daher oft zu Verwirrung.

Am sinnvollsten ist es, die SD oder MMC im Rechner mit einer EXT2-Partition auszurüsten. Dafür eignet sich z.B. die Freeware-Version von Paragon - Partitionmanager (V7.5, nur eine Registrierung nötig, keine Kosten). Damit habe ich es bisher immer geschafft, der Karte Leben einzuhauchen. Auch der häufig zitierte Panasonic SD-Formatter kann hier helfen.

Wichtig ist aber, dass eine Partition vorhanden ist.

Und bitte nehmt Abstand davon, Euch mittels ipkg ext2fsprogs und FDISK-Pakete runterzuladen und das auf dem Router selber anzulegen. Wahrscheinlich hängt er sich auf.

### Alte Methode

Erstmal empfehlen wir, einen Kernel zu kompilieren, der MS-DOS-Partitionen und VFAT unterstützt. Die Unterstützung für Partitionen muss im Kernel eingebaut sein, die für VFAT kann auch als Modul in den Kernel eingebunden werden. Es wäre vielleicht nützlich folgende Dinge ins .config aufzunehmen:

```
CONFIG_PARTITION_ADVANCED=y
CONFIG_MSDOS_PARTITION=y
CONFIG_FAT_FS=y
CONFIG_MSDOS_FS=y
CONFIG_VFAT_FS=y
```

Jetzt besorgt man sich den Treiber und das Makefile. Das Makefile muss angepasst werden, um an die Stelle zu zeigen wo die OpenWRT-Header liegen und wo sich der mipsel-Compiler befindet. Wenn das gemacht wurde, einfach nur noch 'make' eintippen (die Warnungen können ignoriert werden).

Das Modul ist nun einsatzbereit. Vergewissere dich, dass im Kartenleser eine SD-Karte steckt und lade dann das Modul. Prüfe mit 'dmesg', ob soweit alles in Ordnung ist. Wenn das der Fall ist, solltest du hoffentlich einige neue Laufwerke in /dev/mmc/... finden. Folgendes ist ein kleiner Auszug aus der "Unterhaltung" mit dem Router:

```
root@radio:~# ls -al /lib/modules/2.4.20/
drwxr-xr-x  1 root  root          0 Jan  1 00:08 .
drwxr-xr-x  1 root  root          0 Jan  1 00:01 ..
lrwxrwxrwx  1 root  root        28 Jan  1 00:01 et.o -> /rom/lib/modules/2.4.20/et.o
-rw-r--r--  1 root  root    50616 Jan  1 00:02 fat.o
-rw-r--r--  1 root  root   12780 Jan  1 00:08 mmc.o
-rw-r--r--  1 root  root   11244 Jan  1 00:03 msdos.o
-rw-r--r--  1 root  root   19156 Jan  1 00:05 vfat.o
lrwxrwxrwx  1 root  root        28 Jan  1 00:01 wl.o -> /rom/lib/modules/2.4.20/wl.o
root@radio:~# insmod mmc
Using /lib/modules/2.4.20/mmc.o
root@radio:~# dmesg | tail -7
mmc Hardware init
mmc Card init
mmc Card init *1*
mmc Card init *2*
Size = 249856, hardsectsize = 512, sectors = 499712
Partition check:
  mmca: p1
root@radio:~# insmod fat
Using /lib/modules/2.4.20/fat.o
root@radio:~# insmod msdos
```

### Aktuelle Methode

## Hinzufügen\_eines\_SD-Kartenlesers\_an\_den\_Linksys\_WRT54G

```
Using /lib/modules/2.4.20/msdos.o
root@radio:~# mount /dev/mmc/disc0/part1 /mnt -tmsdos
root@radio:~# ls -al /mnt
drwxr-xr-x  2 root    root          16384 Jan  1  1970 .
drwxr-xr-x  1 root    root           0 Jan  1  00:01 ..
-rwxr-xr-x  1 root    root           0 Jan  1  00:07 bossepr0.pic
-rwxr-xr-x  1 root    root        22646 Jan  1  00:02 ld-uclib.so
-rwxr-xr-x  1 root    root         12780 Jan  1  2000 mmc.o
-rwxr-xr-x  1 root    root       1048576 Jan  1  2000 temp.bin
-rwxr-xr-x  1 root    root     16777216 Jan  1  2000 temp2.bin
-rwxr-xr-x  1 root    root     16777216 Jan  1  2000 temp3.bin
-rwxr-xr-x  1 root    root         693 Jan  1  2000 temp4.bin
root@radio:~# df
Filesystem            1k-blocks      Used Available Use% Mounted on
/dev/root              896           896         0 100% /rom
/dev/mtdblock/4       2176          1580         596  73% /
/dev/mmc/disc0/part1 249728        33856      215872   14% /mnt
```

### Etwas Hilfe beim Kernel kompilieren

Der einfachste Weg, um einen lauffähigen Kernel mit der gewünschten Dateisystem-Unterstützung zu erstellen, ist vermutlich, sich OpenWRT runterzuladen und das Flash-Image erstellen. Wenn man sich damit auskennt ist es ziemlich einfach, die Einstellungen für den eigenen Kernel anzupassen. Einfach ins buildroot/build\_mipsel/linux Verzeichnis wechseln und 'make menuconfig' eintippen. Dann nach File Systems -> Partition Types wechseln und 'Advanced partition selection' und 'PC BIOS (MSDOS partition tables) support' anklicken. In 'File systems' sollte ebenfalls 'DOS FAT fs support' aktiviert werden, optional auch noch 'VFAT (Windows 95) fs support'. Dannach die Änderungen speichern und das Menü verlassen. Mit 'make dep zImage' den Kernel erstellen. Zum Schluss nur noch das OpenWRT-Image erstellen und der neue Kernel wird automatisch integriert.

### Welche SD Karten funktionieren?

Da nicht alle SD Karten im Router funktionieren (>2GB geht aus Adressierungsgruenden generell nicht) hier eine Liste (bitte ergaenzen bei wem ne Karte funktionier!!!) von Karten die Funktionieren:

Hersteller	Größe	Seriennummer	Wo gekauft
HAMA	256MB	GH0607008cH0608Pb	Media Markt
SanDisk	512MB	AX0609904179D	Foto Laden
PHOTO PORST	256MB	009700483205	Foto Laden
Transcend 80x	1024MB	6451AE 1G 02DS0	Lidl Dez 06
Transcend	1024MB	8111AB 1G D1DS1	Promarkt Januar 07
Kingston	1024MB	SD-M01G 0646 SM2949U	Media Markt
Kingston	1024MB	SD-M01G 0738 SP95404U	Real,-
SanDisk	1024MB	SDSDB-1024-E10	Saturn Mai 07

## Hinzufügen\_eines\_SD-Kartenlesers\_an\_den\_Linksys\_WRT54G

Kingston	1024MB	SD-M01G-1024-D12SA	Saturn Jan 08
CnMemory	512MB	-	Netto
takeMS 60x	1024MB	S442CC01G EHBB 0650	Marktkauf
MMC Karte	16MB	-	aus Nokia 3650
RS-MMC Karte	32MB	-	aus Nokia 6630
RS-MMC Karte	64MB	-	aus Nokia 6630
Medion	128MB	AK0409VB	??
Lexar MMC	64MB	B01350125	??
RS-MMC Karte	512MB	MMCmobile extre me memory	Birg (?)
takeMS	1024MB	- normal keine 60X -	ebay
Transcend	1GB	6451AE 1G 01DS0	Promarkt
SanDisk Ultra II	2GB	BE0725411366D	-
Transcend	1GB	TS1GSDC SN: 183611-69XX	CO Conrad 29.11.07 7,99?
Apacer	1GB	SD1GBM04S0442	STEG Computer
SanDisk ca. 6-7 EUR	2GB	SDSDB2048E11	Saturn / Conrad / MediaMarkt
Secure Digital digi BRAIN	128MB	-	Penny
Emtec	2GB	S445CC02GAHBE	Real

### SD Karten die nicht funktionieren!

Diese SD Karten wurden getestet, leider ohne jeglichen Erfolg.

Hersteller	Größe	Seriennummer	Wo gekauft
MiniSD + SD Adapter	2048MB	USM2GBPH06C	EBAY / MEMORYWORLD GmbH & Co. KG
Patriot Memory	1GB	PSF1G40SD-PC	PC- Laden
Patriot Secure Digital Card	1024 MB	-	www.Alternate.de
Transcend	1GB	TS1GSDC	Conrad Eletronic Karte wurde am 29.11 erfolgreich installiert
Transcend	1GB	TS1GUSD	30.10.2010 / Conrad Eletronic (MMC-INIT-Error)
Transcend	2GB	TS2GUSD	30.10.2010 / Conrad Eletronic (MMC-INIT-Error)
Toshiba	2GB	0727TQ9026U	Media Markt

Welche SD Karten funktionieren?



## Hinzufügen\_eines\_SD-Kartenlesers\_an\_den\_Linksys\_WRT54G

Kingston	1GB	30964-001.A00LF	Media Markt
Canon	16MB	HB28H016MM2	Canon DigiCam
Traveler	128MB	BQ4HC014280	ALDI Traveler DigiCam 8MPixel

## Fehlerbehebung

Was tun wenn es doch nicht funktioniert?

- GPIO pins durchmessen/checken

```
while true; do gpio disable 2; sleep 1; gpio enable 2; sleep 1; done
```

```
while true; do gpio disable 3; sleep 1; gpio enable 3; sleep 1; done
```

- etc helfen hier sehr, einfach im telnet/ssh interface eingeben, ggf die GPIO nummer aendern.  
Diese befehle schalten im sekudentakt die GPIO Pins an und aus, die blinken dann bzw man Kann messen ob ca 3,35V ankommen oder nicht.
- Kabellänge auf das minimum reduzieren, auf keinen Fall laenger als 20cm, am besten nur so 5 oder so...
- Andere SD Karte probieren

---

## Braucht den Kram hier drunter noch jemand?

*und vor allem: was macht das hier im SD-Karten-Mod?*

## Version herausfinden

Die ersten 4 Stellen der Seriennummer (Zu finden auf der Unterseite des Routers) repräsentieren die Hardware-Version:

```
CDF0 = WRT54G v1.0  
CDF1 = WRT54G v1.1  
CDF3 = WRT54G v1.1  
CDF5 = WRT54G v2.0  
CDF7 = WRT54G v2.2  
CDF8 = WRT54G v3.0  
CDF9 = WRT54G v3.1  
CDFA = WRT54G v4.0
```

```
CGN0 = WRT54GS v1.0  
CGN1 = WRT54GS v1.0  
CGN2 = WRT54GS v1.1  
CGN3 = WRT54GS v2.0  
CGN4 = WRT54GS v2.1  
CGN5 = WRT54GS v3.0  
CGN6 = WRT54GS v4.0
```

## Informationen über Hardware

Infos über die Hardware erhaltet ihr auf folgenden Seiten:

<http://Router-IP/SysInfo.htm> (die Versionsangabe hier ist nicht immer zuverlässig)

<http://Router-IP/Cysaja.asp>

<http://Router-IP/Info.htm> (Um diese Seite als Startseite unter <http://Router-IP/> zu sehen, kann man sie im GUI unter "Administration ->Management ->Remote Router Access -> "Enable Status Site" aktivieren.)

## Overclocking der CPU

Mit folgendem Befehl wird die CPU übertaktet (hier auf 240 MHz):

```
nvramp set clkfreq=240  
nvramp commit
```

Um die aktuelle MHz-Einstellung zu erhalten, folgenden Befehl eingeben:

```
nvramp get clkfreq ODER cat /proc/cpuinfo ODER nvramp show | grep clkfreq
```

Nicht alle Frequenzen sind zulässig! Mit 216 MHz läuft die CPU synchron mit der Onboard-Peripherie und behebt einige Probleme. Weiter Frequenzen: 192, 200, 216, 222, 228, 240, 252, 264, 272, 280, 288 und 300 MHz. Auch bei korrekter Frequenz sind Ausfälle aufgetreten. In diesem Fall hilft nur noch ein Jtag-Interface. Näheres auf <http://wiki.dd-wrt.com/wiki/openwrt.org/>.

Werte ab 264 MHz sollen angeblich keine Auswirkungen haben und in /proc/cpuinfo ist dann immer der selbe BogomIPS-Wert zu sehen. Ein schöner Erfahrungsbericht/HowTo ist [hier](#) zu lesen.